

M2TRec: Metadata-aware Multi-task Transformer for Large-scale and Cold-start free Session-based Recommendations

Walid Shalaby
walid_shalaby@homedepot.com
The Home Depot
Atlanta, Georgia, USA

Sejoon Oh
soh337@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Amir Afsharinejad
amir_afsharinejad@homedepot.com
The Home Depot
Atlanta, Georgia, USA

Srijan Kumar
srijan@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Xiquan Cui
xiquan_cui@homedepot.com
The Home Depot
Atlanta, Georgia, USA

ABSTRACT

Session-based recommender systems (SBRs) have shown superior performance over conventional methods. However, they show limited scalability on large-scale industrial datasets since most models learn one embedding per item. This leads to a large memory requirement (of storing one vector per item) and poor performance on sparse sessions with cold-start or unpopular items. Using one public and one large industrial dataset, we experimentally show that state-of-the-art SBRs have low performance on sparse sessions with sparse items. We propose M2TRec, a Metadata-aware Multi-task Transformer model for session-based recommendations. Our proposed method learns a transformation function from item metadata to embeddings, and is thus, item-ID free (i.e., does not need to learn one embedding per item). It integrates item metadata to learn shared representations of diverse item attributes. During inference, new or unpopular items will be assigned identical representations for the attributes they share with items previously observed during training, and thus will have similar representations with those items, enabling recommendations of even cold-start and sparse items. Additionally, M2TRec is trained in a multi-task setting to predict the next item in the session along with its primary category and subcategories. Our multi-task strategy makes the model converge faster and significantly improves the overall performance. Experimental results show significant performance gains using our proposed approach on sparse items on the two datasets.

CCS CONCEPTS

• Information systems → Recommender systems.

ACM Reference Format:

Walid Shalaby, Sejoon Oh, Amir Afsharinejad, Srijan Kumar, and Xiquan Cui. 2022. M2TRec: Metadata-aware Multi-task Transformer for Large-scale and Cold-start free Session-based Recommendations. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22)*, September 18–23, 2022, Seattle, WA, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys '22, September 18–23, 2022, Seattle, WA, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9278-5/22/09...\$15.00

<https://doi.org/10.1145/3523227.3551477>

Seattle, WA, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3523227.3551477>

1 INTRODUCTION

Session-based recommender systems (SBRs) accurately model sequential and evolving preferences of users from their session data (e.g., clicks and add-to-cart events). Session data can be associated with item metadata, allowing SBRs to capture item dependencies at the attribute level within the session. However, most of the existing SBRs take the IDs of users and items as the main input source to learn session contexts and produce next item recommendations [13, 14, 40]. Recent hybrid models demonstrated improved performance when combining item embeddings and their attributes to be used as additional side information [5, 10, 27, 31, 32, 32, 38].

However, there are two major issues that these recommendation models face. The first issue is that they cannot scale with the gigantic sizes of industrial datasets. For instance, The Home Depot (THD) industrial dataset used in this paper has approximately 40 million sessions and 0.6 million items. Since the dataset is created by sampling several months of online sessions, the actual full dataset (e.g., a year-long one) will be even much bigger. Most SBRs [5, 8, 13, 14, 19, 40, 41, 44] that utilize a large item-ID embedding matrix can suffer from slow training or memory shortage problems.

The second issue arises due to cold-start items and sparse sessions, i.e., sessions that contain new or unpopular items. SBRs will have limited or no ability to generate good representations for such items since they have no-to-few interactions. Moreover, many existing models are incapable of scoring and recommending new items unseen during training [25, 35]. Even combining metadata information with item-IDs, i.e., item embeddings, to learn compound item representation results in only a slight performance improvement compared to using item-ID only [5, 10, 16, 27, 31, 33, 38]. This can be attributed to the model overfitting item-ID as the main feature.

To tackle the above issues, we propose M2TRec, a Metadata-aware Multi-task Transformer model (Figure 1). M2TRec is completely item-ID free (i.e., no item-ID embeddings) and uses only item attributes such as title, category, brand, color, and other metadata to learn item representations. Since M2TRec does *not* require creating and learning a large item-ID embedding matrix, it can be easily applied to large industrial datasets. In addition, new items will still have accurate representations using their metadata attributes, and

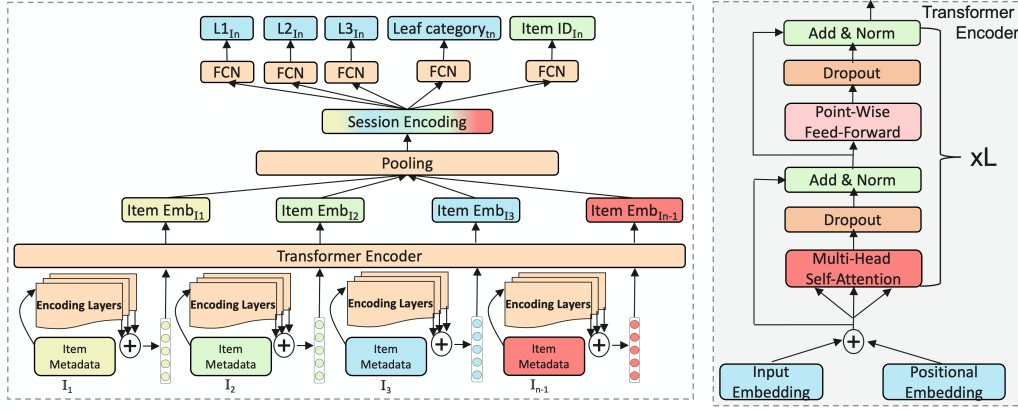


Figure 1: M2TRec Architecture. (left) M2TRec first encodes metadata of each item in a session into embeddings and feeds the concatenation of those metadata embeddings to a Transformer encoder. A current session encoding is obtained via average-pooling of the item encodings from Transformer, and the session encoding is used for next item and next category predictions. (right) Detailed architecture illustration of a Transformer encoder used in M2TRec.

their similarity with previously-observed items can be captured even if they have no or few interactions with users, making it suitable even in highly dynamic deployment settings where the item catalog changes frequently. Finally, recent research has shown that multi-task learning (MTL) results in increased performance and generalizability of the model [6, 7]. Thus, given the focus on leveraging item metadata, we design M2TRec as a multi-task SBRs trained not only to predict the next item but also its category and subcategories to enhance the prediction accuracy of an individual task.

Via thorough experiments, we demonstrate the superior performance of the proposed model on real-world datasets. Our multi-task training allows faster convergence, higher accuracy with fewer iterations, and robust performance with fewer training data (some experimental results are omitted due to space limits). Our scalable architecture serves both item and category recommendations in one model with higher prediction performance than baselines in both tasks.

2 RELATED WORK

Neural networks have served as the main architecture of existing SBRs. Earlier works [8–10] utilize a recurrent neural network (RNN) to model sequential dependencies of items within a session. Recently, attention-based approaches [13, 14, 19, 23, 37] and graph neural network (GNN)-based methods [39–42, 44] have been proposed to enhance the longer and deeper dependencies for SBRs. Furthermore, Transformer [34]-based SBRs [2, 4, 5, 18] show the state-of-the-art prediction performance due to its powerful and efficient self-attention mechanism. Multi-task learning (MTL) [11, 16, 20] has been also adopted for SBRs to enhance the next item prediction via generalization. However, the above models have shortcomings that they (1) are susceptible to cold-start items or sessions, (2) cannot predict the categories of the next item in a session, or (3) are not scalable to the real-world billion-scale recommendation setting since they have to store item-IDs and their embeddings, which are this paper is solving.

Methodologies are developed to incorporate item metadata into SBRs for modeling user/item dependencies [5, 10, 12, 16, 21, 27, 28,

31, 33, 38, 43]. However, the vast majority of such methodologies have at least one of the following two shortcomings. The first shortcoming is that cold-start items and/or users are removed during the pre-processing of datasets used for evaluations of proposed models [5, 12, 21, 28, 43]. Item metadata alone cannot give the model the ability to recommend such cold-start items [17]. The model needs to have a separate mechanism for using items' content information and representation to recommend cold-start items. Tagliabue et al. [30] propose a pipeline to learn accurate cold-start item representations with small changes to an existing model infrastructure. However, a separate neural model needs to be trained to obtain the cold-start embeddings, which limits the scalability of the solution. Raziperchikolaei et al. [22] and Zheng et al. [45] suggest hybrid and metadata-aware recommendation models to predict implicit feedback for cold-start items of users, respectively. However, those models are not designed for the session-based recommendation setting. The second shortcoming of such methodologies is that they do not make use of item titles or descriptions as attribute features for capturing product similarities [12, 16, 21, 27, 33, 38]. In such cases, item-IDs are used as inputs to represent different products. Such representation is unable to incorporate any relevant information about cold-start items as opposed to using title encodings.

3 METHODOLOGY

Next Item Prediction: We denote a user session $\mathcal{S} = [I_1, I_2, I_3, \dots, I_n]$ as a sequence of items a user interacted within that session. Each item $I_k = \{A_{k,1}, A_{k,2}, A_{k,3}, \dots, A_{k,m}\}$ is described by a set of m attributes which could be context-specific or item-specific. In this work, we consider item-specific attributes only (e.g., title, description, category). Each attribute A could be either textual, categorical, or numerical. In the setting of session-based recommendations, we are given a session \mathcal{S} , and our objective is to maximize the prediction probability of the next item the user is most likely to interact with given all previous items in \mathcal{S} . Formally, the probability of the target item I_n can be formulated as:

$$p(I_n | \mathcal{S}_{[I_{<n}]}; \theta) \quad (1)$$

where θ denotes the model parameters and $\mathcal{S}_{[I_{<n}]}$ denotes the sequence of items prior to the target item I_n . As in previous works [9,

Table 1: Statistics of datasets used in the experiments.

Dataset	Diginetica	THD
# of training and test sessions	191K, 16K	39M, 1.5M
# of items	119K	575K
Metadata / Attributes	Product title, Category	Product title, Categories (L1, L2, L3, Leaf), Manufacturer, Brand, Department Name, Class Name, Color
Prediction tasks	Item-ID, Category	Item-ID, Categories (L1, L2, L3, Leaf)

13, 14, 40], we generate dense next item sub-sequences from each session S for training and testing. Therefore, a session S with n items will be broken down into $n - 1$ sub-sequences such as $\{([I_1], I_2), \dots, ([I_1, I_2, \dots, I_{n-1}], I_n)\}$, where $([X], Y)$ means X as the input sequence of items and Y as the target next item.

Item Metadata Encoding: Item metadata can be numerical, categorical, or unstructured such as title, description, and image. We propose a unified method for representing all item attributes. The objective is to map every attribute A into a real-valued vector $v_A \in \mathbb{R}^{d_A}$. Numerical attributes r are represented as a single-valued vector $v_r \in \mathbb{R}$. Categorical attributes $C \in \{c_1, c_2, \dots, c_s\}$ are encoded into vectors v_C using an embedding layer dedicated to each attribute, i.e.,

$$v_C = c_i \theta^{(C)} \in \mathbb{R}^{d_C} \quad (2)$$

where c_i is the one-hot encoded value of C , $\theta^{(C)} \in \mathbb{R}^{s \times d_C}$ are the weights of the category embedding matrix, s is the number of possible values of C , and d_C is the dimensionality of C 's vector.

Textual attributes T are first tokenized using a subword tokenizer [26] to obtain individual tokens $[\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_t]$ and then encoded into vectors v_T . A simple and efficient encoding strategy is to create a dedicated embedding layer for T to map each token \mathbf{w} into a vector and then aggregate the token vectors using mean or max pooling, i.e.,

$$v_T = \text{Pool}_{i=1}^t (w_i \theta^{(T)}) \in \mathbb{R}^{d_T} \quad (3)$$

where w_i is the one-hot encoded value of token \mathbf{w}_i , $\theta^{(T)} \in \mathbb{R}^{k \times d_T}$ are the weights of the token embedding matrix, k is vocabulary size of T , and d_T is the dimensionality of T 's vector. Further enhancements to textual attributes encoding can be achieved by sharing the encoding parameters across all textual attributes that have similar vocabularies such as item title, description, category, color, etc. Although the weight-sharing scheme is expected to reduce the training time, it may increase the overall model size. This is because the vector size would be the same for all the attributes that share the same encoder regardless of their vocabulary size. This will lead to high memory and storage requirements when deploying the model in production. Alternatively, in this work, we use a separate embedding layer for each textual attribute as in Eq. (3) and choose its vector size proportional to the attribute's vocabulary size.

Session Encoding: After encoding all metadata features for an item I_k at position k in the input session S , we concatenate all the feature vectors to create a compound vector representation v_{I_k} for I_k , i.e.,

$$v_{I_k} = \text{concat}(v_{A_1}, v_{A_2}, \dots, v_{A_m}) \in \mathbb{R}^{d_I} \quad (4)$$

where d_I is the summation of the lengths of all feature vectors. Note that item-IDs are not used to create the compound representation v_{I_k} in Equation (4). We then use the compound representations of items in S as input to the session encoder in a pre-fusion fashion to learn a session encoding v_S . First, Transformer encoder [34] generates contextual encodings for each session item v_{I_k} , followed

by an average-pooling layer to generate the session encoding v_S , i.e.,

$$v_S = \text{Pool}_{k=1}^{n-1} (\text{Trans-Enc}(v_{I_k}; \theta^{(encs)})) \in \mathbb{R}^{d_S} \quad (5)$$

where $\theta^{(encs)}$ is the model parameters of the Transformer encoder trained with sessions S .

Multi-task Learning: M2TRec incorporates multi-task learning (MTL) to boost the performance of next item prediction. MTL has proven to be an effective mechanism to reduce the risk of overfitting, learn more generalized shared representations for all the tasks, and improve the overall performance on each task by sharing the knowledge acquired from other related tasks [24]. The target space of next item prediction (i.e., all item-IDs) is much larger than the space of other item attributes such as all categories or brands. Therefore, the task of next item category or brand prediction should be easier to learn than next item prediction. Moreover, learning such auxiliary tasks would benefit the task of next item prediction since it biases the metadata encoding and the Transformer encoder layers to learn representations that are close not only to next item, but also to other similar items belonging to the same category or brand, thus, narrowing down the space of possible next item candidates to a much smaller set of items. To this end, we train M2TRec to predict next item attributes (e.g., item categories) as auxiliary tasks to the task of next item-ID prediction. For each task, including next item-ID prediction, we create a prediction head composed of Fully Connected Layer (FCN) followed by Softmax function to generate the probability distribution over all candidates for the corresponding task, i.e.,

$$\hat{y}_k = \text{softmax}(\text{FCN}_k(v_S; \theta^{(k)})) \in \mathbb{R}^{d_k} \quad (6)$$

where $\theta^{(k)}$ is the parameters of the FCN for the k^{th} task, and d_k is the total number of possible outputs of the k^{th} task (e.g., total number of categories for category predictions).

The loss of the k^{th} task prediction head and overall prediction of M2TRec are calculated using cross-entropy loss as follows, respectively:

$$\mathcal{L}_k = - \sum_{i=1}^{d_k} y_{k_i} \cdot \log \hat{y}_{k_i}, \quad \mathcal{L} = \sum_{k=1}^N \mathcal{L}_k \quad (7)$$

where y_k is a one-hot encoding including the ground-truth information for the k^{th} task.

4 EXPERIMENTS

Datasets: We conducted our experiments on two real-world datasets. We exclude all sessions with only one item. Table 1 shows dataset statistics along with the item metadata we used and the prediction tasks for each dataset.

Table 2: The performance of M2TRec on next item prediction task on all sessions (All) and sessions with tail items (Sparse) compared to other baseline methods. Tail items indicate items with less than 10 occurrences in a dataset. (Bold indicates the best model, while the second-best model is underlined).

Dataset	Diginetica						THD					
	HIT@20		Recall@20		MRR@20		HIT@20		Recall@20		MRR@20	
	All	Sparse	All	Sparse	All	Sparse	All	Sparse	All	Sparse	All	Sparse
STAMP	25.45	16.05	41.61	28.89	7.17	4.44	31.91	14.65	<u>38.49</u>	18.90	14.88	6.46
GRU4Rec	29.09	<u>22.33</u>	45.45	36.14	<u>8.51</u>	<u>6.45</u>	<u>32.27</u>	<u>16.81</u>	37.64	<u>20.28</u>	<u>14.97</u>	7.93
NARM	30.49	21.78	<u>47.60</u>	<u>36.90</u>	8.13	5.55	28.98	11.31	35.45	15.59	11.63	3.97
M2TRec	35.47	25.51	53.70	41.01	9.75	6.63	34.78	19.05	41.51	24.86	15.65	7.38
Improvement	16.33%	14.2%	12.82%	11.1%	14.57%	2.79%	7.78%	13.3%	7.85%	22.6%	4.54%	-6.94%

- *Diginetica*¹ is an E-commerce dataset that was a part of CIKM Cup 2016 challenge. We use the transactional and product data and use pre-processing similar to [40].
- *THD* is an E-commerce dataset obtained from The Home Depot, the largest home improvement retailer in the USA. The dataset is composed of Add-to-Cart (ATC) events within millions of online sessions. Similar to SIGIR 2021 data challenge dataset[29], training data is created by sampling several months of online purchase sessions. Test data is sampled from a disjoint and adjacent time period. The dataset has rich product metadata including 7 attributes: product title, categories (L1, L2, L3, Leaf), brand, manufacturer, color, department, and class name.

Baselines: We select the following 3 state-of-the-art SBRSs to compare them with M2TRec: (1) *GRU4Rec* [9]: A popular and first-generation SBRS that utilizes a Gated Recurrent Unit (GRU) [3] to model long-term dependencies within a session, (2) *NARM* [13]: An attention-based SBRS that employs a hybrid encoder to reflect a user’s global and local interests with an attention mechanism, and (3) *STAMP* [14]: An attention/memory-based SBRS that incorporates a user’s short-term and long-term interests via a short-term attention and long-term memory modules, respectively.

Evaluation Metrics: We use HIT@K, Recall@K, and MRR@K [36] to evaluate the performance of M2TRec. All the metrics range from 0 to 1, and higher values are better. We choose $K = 20$ since it is a standard value [15].

Implementations: We used open-source implementations for all baseline methods². With M2Trec, we encode all the attributes as textual. We use a dedicated embedding layer for each attribute followed by average pooling of individual tokens’ vectors. The embedding dimension is set proportionally to the total number of distinct tokens of the corresponding attribute vocabulary. For the Transformer encoder, we used 2 encoder layers with 8 attention heads in each layer and point-wise feed-forward networks consisting of two fully-connected layers [2048, 128] with a ReLU activation [1] in between. We fine-tuned all the hyperparameters of M2TRec on a validation dataset sampled randomly from THD data.

Next item prediction task on all sessions: The performance of all the models on all sessions of the two datasets is shown in Table 2. As we can notice, M2TRec outperforms all the baselines across all the evaluation metrics. On Diginetica, the relative performance improvements of HIT@20, Recall@20, and MRR@20 are in the

range of 13% ~ 16%. On THD dataset, the relative performance improvements are in the range of 5% ~ 8%. As in [27], we compute the relative performance improvement of a metric as the difference in the performance of M2TRec and the second runner over the performance of the second runner on that metric reported in percentage. These improvements indicate the effectiveness of utilizing item metadata and the multi-task learning regime which are unique to M2TRec, compared to other baselines which use only item-ID as the main and only input for session-based recommendations.

Next item prediction task on sparse sessions: Table 2 highlights the performance of all the models on sparse sessions containing cold-start or tail items in the two datasets. Tail items indicate items with less than 10 occurrences in a dataset. These sessions represent 34% and 12% of the total sessions in Diginetica and THD datasets respectively. As we can notice, M2TRec relative improvements on sparse sessions are much higher than all the other models, especially on HIT@20 and Recall@20 for both datasets (e.g., 11% ~ 23% boost on both datasets). These results demonstrate the effectiveness of our proposed item-ID free approach on sparse sessions and its robustness in mapping tail and cold-start items within these sessions into meaningful representations based on their metadata.

Predicting next item’s category: One of the main objectives of this research is to develop a scalable architecture that serves both item and category recommendations in one model using an efficient MTL regime. We found significant performance gains when jointly training our model to predict next item and its categories at different levels of the catalog taxonomy (see ablation study below). We demonstrate the efficacy of training our SBRS to predict next category over deriving it from session items by comparing the category prediction performance against two heuristics: (1) *Personalized top-N Frequent*: This simple heuristic uses past session items’ categories and recommends the most frequent ones, and (2) *Top-N Predicted*: This simple strategy works by first predicting top-N next items from a metadata-aware single task model called MeTRec (see ablation study below), and then uses their categories as recommendations such that the category of the top-ranked next item will be ranked first and so on.

Figure 2 shows the performance of category recommendation using M2TRec against the two heuristics. Performance is measured in terms of HIT@20 and reported for L1, L2, L3, and leaf categories of THD dataset. As we can notice, the performance of M2TRec is significantly better than the two other strategies across all tasks. For example, on leaf category prediction, the performance gains are in the range of 3%~45%. These results demonstrate that M2TRec can effectively perform next category prediction tasks.

¹<https://competitions.codalab.org/competitions/11161>

²<https://github.com/rn5l/session-rec>

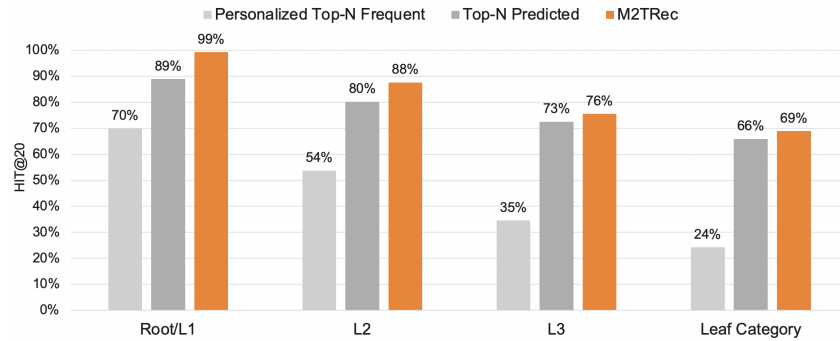


Figure 2: M2TRec Performance on Category Recommendation

Table 3: The performance of M2TRec on all sessions (All) and sessions with tail items (Sparse) compared to variants of M2TRec.

Dataset	Diginetica						THD						
Method	HIT@20		Recall@20		MRR@20		HIT@20		Recall@20		MRR@20		Model
	All	Sparse	All	Sparse	All	Sparse	All	Sparse	All	Sparse	All	Sparse	Size
TRec _{id}	30.24	16.93	46.96	29.45	8.55	4.35	32.90	11.58	39.40	15.22	15.31	4.90	110.3M
MuTRec _{id}	32.69	21.10	50.13	35.18	<u>9.50</u>	5.71	<u>33.73</u>	13.93	<u>40.35</u>	17.92	<u>15.43</u>	5.90	111.1M
TRec _{title}	32.17	20.41	49.40	34.64	8.70	5.22	32.29	16.48	38.64	21.19	14.25	6.32	56.1M
MeTRec	<u>34.78</u>	<u>24.99</u>	<u>52.50</u>	<u>40.43</u>	9.33	<u>6.22</u>	<u>33.54</u>	<u>17.10</u>	40.04	<u>21.97</u>	15.17	<u>6.42</u>	86.5M
M2TRec	35.47	25.51	53.70	41.01	9.75	6.63	34.78	19.05	41.51	24.86	15.65	7.38	89.5M

Ablation Study of M2TRec: To investigate the contribution of each component of M2TRec, we developed the following variants: (1) ***TRec_{title}***: A variant of M2TRec which uses item title as the only input feature without any additional meta-data. It leverages the same architecture in Figure 1, but has one prediction head only to predict next item-ID, (2) ***MeTRec***: Metadata-aware variant which utilizes all metadata as input features. This variant also has one prediction head to predict next item-ID, (3) ***TRec_{id}***: A variant of M2TRec which uses item-IDs as the only input feature without any additional meta-data attributes. It leverages the same architecture in Figure 1 but has only one prediction head to predict next item-ID, and (4) ***MuTRec_{id}***: Multi-task variant of *TRec_{id}*. The model is trained on the same tasks as M2TRec, but uses only item-IDs as the input features (i.e., it does not use other metadata features).

The performance of M2TRec and its variants on all and sparse sessions is shown in Table 3, where model size indicates the number of model parameters. M2TRec outperforms all other variants significantly, especially on sparse sessions where the performance gains on Diginetica dataset are in the range of 1%~9% in terms of HIT@20 and 1%~12% in terms of Recall@20. On THD dataset, performance gains are in the range of 2%~7% and 3%~10% in terms of HIT@20 and Recall@20 respectively. As we include all metadata in MeTRec, the performance on sparse sessions outpaces all other variants. Moreover, the performance on all sessions improves significantly and outpaces TRec_{id} and its multi-task version (MuTRec_{id}) on Diginetica, while it is on par with MuTRec_{id} on THD dataset. This demonstrates the usefulness of metadata-awareness and its sufficiency in providing competitive performance to classical item-ID based SBRs. As we can notice, M2TRec and MeTRec are about 19%-22% less in size than the item-ID based variants. Besides, all the metadata-aware variants are more scalable to the increase in item catalog size compared to the item-ID based variants.

5 CONCLUSION

This work provides a scalable and practical solution for leveraging metadata to learn from cold-start items in the recommendation process. The key is using an item-ID free approach for recommendations. By using a metadata-based representation of items, the M2TRec model learns the representation for items with zero or few interactions. Through experiments on two datasets, we show that M2TRec outperforms several state-of-the-art session-based recommendation models. Multi-task learning contributes to the model’s predictive performance. Importantly, M2TRec’s core ideas help in generating fast and accurate recommendations for cold start-items, sessions with tail items, and for the task of category prediction.

REFERENCES

- [1] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* (2018).
- [2] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2597–2601.
- [3] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. 103–111.
- [4] Gabriel de Souza Pereira Moreira, Sara Rabhi, Ronay Ak, and Benedikt Schifferer. 2021. End-to-End Session-Based Recommendation on GPU. In *Fifteenth ACM Conference on Recommender Systems*. 831–833.
- [5] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential/Session-Based Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 143–153.
- [6] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 1554–1557.
- [7] Guy Hadash, Oren Sar Shalom, and Rita Osadchy. 2018. Rank and rate: multi-task learning for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 451–454.
- [8] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

- [9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [10] Balázs Hidasi, Massimo Quadroni, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 241–248.
- [11] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*.
- [12] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* 27 (12 2017). <https://doi.org/10.1007/s11257-017-9194-1>
- [13] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.
- [14] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. , 1831–1839 pages.
- [15] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM conference on recommender systems*. 462–466.
- [16] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1091–1100.
- [17] Gabriel Moreira, Dietmar Jannach, and Adilson Cunha. 2019. On the Importance of News Content Representation in Hybrid Neural Session-based Recommender Systems. *arXiv preprint arXiv:1907.07629v3* (07 2019).
- [18] Gabriel de Souza P Moreira, Sara Rabhi, Ronay Ak, Md Yasin Kabir, and Even Oldridge. 2021. Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. *arXiv preprint arXiv:2107.05124* (2021).
- [19] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. 2020. An intent-guided collaborative machine for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1833–1836.
- [20] Nan Qiu, BoYu Gao, Feiran Huang, Huawei Tu, and Weiqi Luo. 2021. Incorporating Global Context into Multi-task Learning for Session-Based Recommendation. In *International Conference on Knowledge Science, Engineering and Management*. Springer, 627–638.
- [21] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting Cross-Session Information for Session-Based Recommendation with Graph Neural Networks. *ACM Trans. Inf. Syst.* 38, 3, Article 22 (may 2020), 23 pages. <https://doi.org/10.1145/3382764>
- [22] Ramin Raziperchikolaei, Guannan Liang, and Young-joo Chung. 2021. Shared Neural Item Representations for Completely Cold Start Problem. In *Fifteenth ACM Conference on Recommender Systems*. 422–431.
- [23] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.
- [24] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).
- [25] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*. 89–96.
- [26] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* (2015).
- [27] Jiayu Song, Jiajie Xu, Rui Zhou, Lu Chen, Jianxin Li, and Chengfei Liu. 2021. CBML: A Cluster-based Meta-learning Model for Session-based Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1713–1722.
- [28] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 909–912.
- [29] Jacopo Tagliabue, Ciro Greco, Jean-François Roy, Bingqing Yu, Patrick John Chia, Federico Bianchi, and Giovanni Cassani. 2021. Sigir 2021 e-commerce workshop data challenge. *arXiv preprint arXiv:2104.09423* (2021).
- [30] Jacopo Tagliabue, Bingqing Yu, and Federico Bianchi. 2020. *The Embeddings That Came in From the Cold: Improving Vectors for New and Rare Products with Content-Based Inference*. Association for Computing Machinery, New York, NY, USA, 577–578. <https://doi.org/10.1145/3383313.3411477>
- [31] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the eleventh ACM conference on recommender systems*. 138–146.
- [32] Bartłomiej Twardowski. 2016. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 273–276.
- [33] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) (RecSys '16). Association for Computing Machinery, New York, NY, USA, 225–232. <https://doi.org/10.1145/2959100.2959160>
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [35] Michail Vlachos, Celestine Dünner, Reinhard Heckel, Vassilios G Vassiliadis, Thomas Parnell, and Kubilay Atasü. 2018. Addressing interpretability and cold-start in matrix factorization for recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 31, 7 (2018), 1253–1266.
- [36] Ellen M Voorhees et al. 1999. The trec-8 question answering track report.. In *Text Retrieval Conference*, Vol. 99. 77–82.
- [37] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.
- [38] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 285–302.
- [39] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 169–178.
- [40] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.
- [41] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. In *30th ACM International Conference on Information and Knowledge Management (CIKM 2021)*. ACM.
- [42] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.
- [43] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenberg, and Jure Leskovec. 2019. Hierarchical temporal convolutional networks for dynamic recommender systems. In *The world wide web conference*. 2236–2246.
- [44] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1921–1924.
- [45] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2021. Cold-start sequential recommendation via meta learner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4706–4713.